CS206 Data Structure

Homework #4

Sungwon Cho

0. About Programs

Program Language: C# 5.0 (.NET 4.5)
Libraries: Standard library of .NET(ex: System, System.Core)
Requirements: OS that supports .NET Framework 4.5 or Mono

1. Number of Digit

1: *int* **procedure** NumDigit(*int* n)
2:   **if** $(1 \leq n < 10)$
3:     **return** 1
4:   **else**
5:     **return** NumDigit($\lfloor n/10 \rfloor$) + 1

Theorem 1. *The procedure NumDigit correctly returns number of digit of positive integer n.*

Proof.
(1) Use mathematical induction on $\lfloor log_{10}n \rfloor$
(2) When $\lfloor log_{10}n \rfloor = 0$ (i.e $1 \leq n \leq 9$) NumDigit is correct(induction step).
 ($\uparrow$ 2.1) Trivial by line 2 and 3.
(3) Assume that NumDigit is correct when $\lfloor log_{10}n \rfloor = k$ (induction hypo).
(4) Then, NumDigit is also correct when $\lfloor log_{10}n \rfloor = k + 1$(induction step).
 ($\uparrow$ 4.1) $\lfloor log_{10}n \rfloor = k + 1 \Leftrightarrow 10^{k+1} \leq n < 10^{k+2}$
 ($\uparrow$ 4.2) In NumDigit(n), it will call NumDigit(n') where $n' = \lfloor n/10 \rfloor$.
 ($\uparrow$ 4.3) Since $10^k \leq n' < 10^{k+1}$, NumDigit gives correct answer for n'.
 ($\uparrow$ 4.4) So, NumDigit(n') + 1 is number of digit of n.
 ($\uparrow$ 4.5) Therefore, NumDigit gives correct answer for n.
(5) $\therefore$ NumDigit is correct for all positive integers. $\square$

2. Importance of Algorithm Order

Effect of processor speed is constant for any input data size n. However, effect of algorithm order is depend on input data size. So generally, if n is very large, effect of algorithm order becomes very huge while effect of processor speed is very small. For example, assume that computer A executes 10^6 assembly commands in 1 sec and computer B executes 10^8 commands in 1 sec. Then, for any input size n, computer A is 100 times slower than computer B. However, assume that order of algorithm X is n^2 and order of algorithm Y is n. If n is bigger than 10^3, algorithm X is 1000 times slower than algorithm Y.

3. Big-O Table

Following is best possible big-O notations and proofs of each items.
(1) $10n = O(n)$
  ($\uparrow$ 1.1) $\forall_{n \geq 0}\, 10n \leq 10 \times n$
(2) $2n^2 = O(n^2)$
  ($\uparrow$ 2.1) $\forall_{n \geq 0}\, 2n^2 \leq 2 \times n^2$
(3) $3logn = O(logn)$ (base of log in big-O and below proof is 2)
  ($\uparrow$ 3.1) $\forall_{n > 0}\, 3logn \leq 3 \times logn$
(4) $2n^2 + 10n = O(n^2)$
  ($\uparrow$ 4.1) $\forall_{n \geq 1}\, 2n^2 + 10n \leq 2n^2 + 10n^2 \leq 12 \times n^2$

4. Big-O #1

$1 + 2 + 3 + \cdots + n = O(n^2)$
($\uparrow$ 1) $\Sigma_{i=1}^{n} i = \frac{1}{2}n^2 + \frac{1}{2}n$
($\uparrow$ 2) $\forall_{n \geq 1}\, \frac{1}{2}n^2 + \frac{1}{2}n \leq \frac{1}{2}n^2 + \frac{1}{2}n^2 \leq 1 \times n^2$

5. Big-O #2

$n^2 + 35n + 6 = O(n^2)$
($\uparrow$ 1) $\forall_{n \geq 1}\, n^2 + 35n + 6 \leq n^2 + 35n^2 + 6n^2 \leq 42 \times n^2$

6. Time Complexity Analysis

```
1:  while (n > 0)
2:      n = n/10
3:  end while
```

Let $n = a_0 a_1 \cdots a_k$ where $a_0 \neq 0$ before execution of any code. And let n_i is the value of n in line 1 after i th while loop execution(so $n_0 = n$). Since n is integer variable, $\forall_{i \geq 0}\, n_{i+1} = \lfloor n_i/10 \rfloor$. Therefore $\forall_{0 \leq i \leq k}\, n_i = a_0 a_1 \cdots a_{k-i}$. So, after $k+1$ th iteration, n in line 1 will be $n_{k+1} = 0$ and program will terminated. Since $k = \lceil log_{10}(n+1) \rceil$, total $2\lceil log_{10}(n+1) \rceil + 1$ operations executed. In big-O notation, its time complexity is $O(logn)$ since $\forall_{n \geq 1}\, 2\lceil log_{10}(n+1) \rceil + 1 \leq 2log_{10}(n+1) + 3 \leq \frac{2}{log10} logn + 5 \leq 1000 \times logn$.

7. Big-O #3

$(n-2)(n-4) = O(n^2)$ $(= O(n) \times O(n))$
($\uparrow$ 1) $(n-2)(n-4) = n^2 - 6n + 8$
($\uparrow$ 2) $\forall_{n \geq 1}\, n^2 - 6n + 8 \leq n^2 + 0 + 8n^2 \leq 9 \times n^2$

8. Tower of Hanoi

Source Files: Program.cs
Instruction: Program.inst.txt
Sample Input, Output: Program.sample.txt